COMP3111 Group Project: Activity 1

October 23, 2024

1 work distribution

Project Group Information										
Group No.	37									
Team Repo	https://github.com/JohnnySy-dio/COMP3111_Group37_Project									
Members						Participation in Activity 1				
Name	GitHub User Id.	Ust Email	Is Team Leader?	Owner of Team Repo?	Role/Task in (Lab5 or Group Project) Create/Manage Team Repo,	Class Diagram	Use-Case Diagram	Use-Case Specifications for task1	Use-Case Specifications for task2	Use-Case Specifications for task3
SY, Ming Lung Johnny	JohnnySy-dio	mlsy	Yes	Yes	editting source	Yes	Yes	No	Yes	No
Li, Haoyuan Danny	Neckistic	hliez	No	No	Editing source	Yes	Yes	Yes	No	No
Zhu, FeiHao	yugeng58	fhzhu	No	No	Editing source	Yes	Yes	No	NO	Yes

Figure 1: Work Distribution

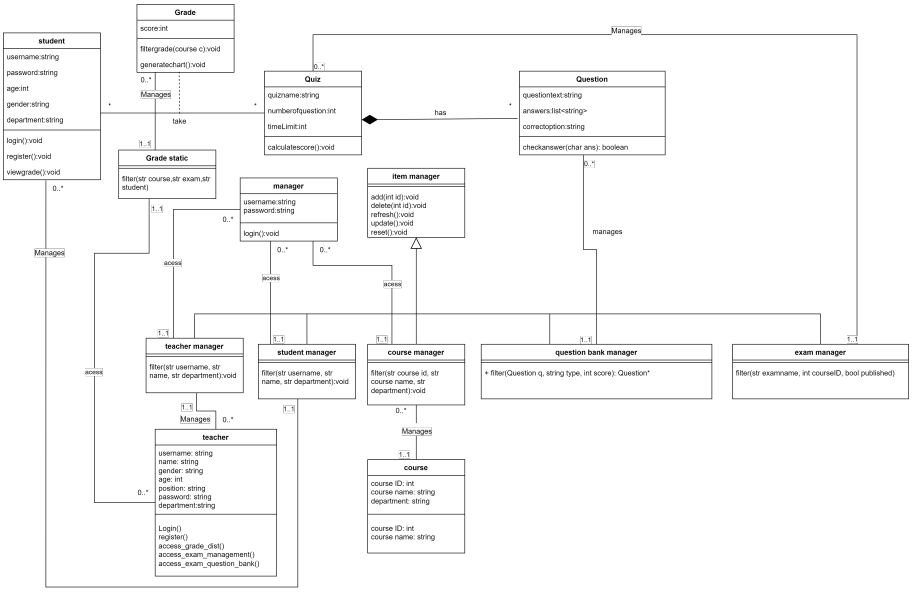


Figure 2: Class diagram

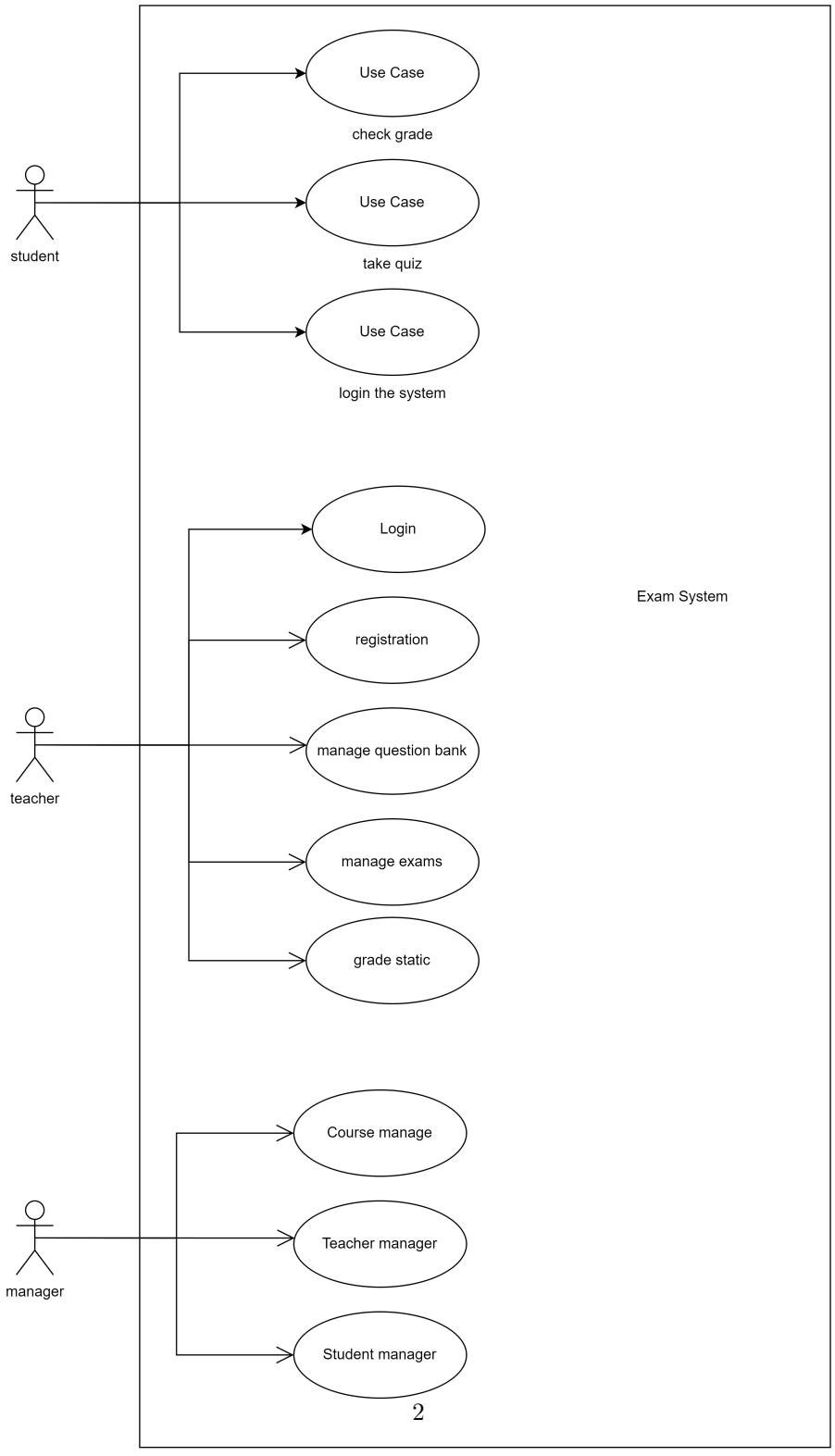


Figure 3: Use-Case Diagram

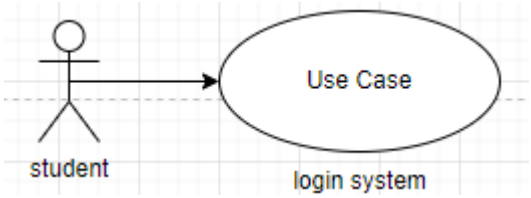
Use-Case Specifications for Task1

Use case: login the system

Brief description

This Use case describes how student login or register the HKUST examination system.

Use-case diagram



Basic Flow

1. The use case begins when the student opens the application.
2. The system displays the login interface. It would show three options which are student login, teacher login and manager login.
3. The student selects student login.
4. The system displays the interface for student to enter login information.
   4.1. If the student has registered on the system
   {Logging In}
   4.1.1. The student enters his own username and password and select login
   4.2. If the student has not registered on the system
   4.2.1. The student selects register
   {Register}
   4.2.2. The system displays the interface for student to enter basic information including Username, Name, Gender, Age, Department and Password
   4.2.3. The student enters the required information
   4.2.4. The system stores the student's information in the database for login.
5. The use case ends.

Alternative Flows

A1: Invalid Login

At {Logging In}, if the password does not match the student's password stored in the database

1. The system notifies the student that the username or the password is incorrect.
2. The flow of events is resumed at {Logging In}.

A2: Username Not Found

At {Logging In}, if the entered username does not exist in the database,

1. The system notifies the student that the username is not found.
2. The flow of events is resumed at {Logging In}.

A3: Registration with Existing Username

At {Register}, if the student enters a username that is already stored I the database,

1. The system notifies the student that the username is already in use.
2. The flow of events is resumed at {Register}.

A4: Empty Fields

At {Logging In} or {Register}, if the student log in or register without entering required fields,

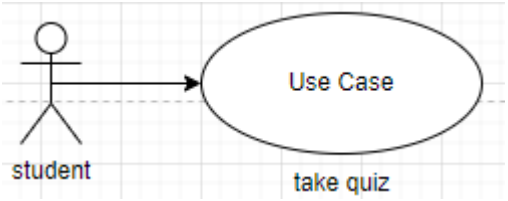
1. The system notifies the student to fill in all required fields.
2. The flow of events is resumed at {Logging In} or {Register}.

Use Case: Take Quiz

Brief Description

This use case describes how a student takes a quiz after logging in.

Use-case Diagram



Basic Flow

1. The use case begins when the student chooses a quiz to take after logging in the system.
2. The system displays the available quizzes.

{Select Quiz}

3. The student selects the desired quiz and clicks start.

{Quiz Begins}

4. The system retrieves and displays the quiz information including quiz name, number of questions, questions and timer.

{Completing Quiz}

5. The student answers the questions.

{Submit Quiz}

6. The student submits the quiz.

{Quiz Ends}

7. The system evaluates the quiz and displays the results.
8. The use case ends.

Alternative Flows

A1: Time Run Out

At any point between {Quiz Begins} and {Quiz Ends}, if the timer expires,

1. The system notifies the student that time's up for the exam.
2. The flow of events is resumed at {Quiz Ends}.

A2: Submit Incomplete Quiz

At {Submit Quiz}, if the student tries to submit the quiz without answering all required questions,

1. The system notifies the student that he has not answer all the questions.
2. The flow of events is resumed at {Completing Quiz}.

A3: Not Selecting Quizzes

At {Select Quiz}, if the student clicks start without selecting any quizzes.

1. The system notifies the student to select a quiz.
2. The flow of events is resumed at {Select Quiz}.

A4: Selecting Completed Quizzes

At {Select Quiz}, if the student selects the quiz he has completed before,

1. The system notifies the student to select a quiz that he has not completed it.
2. The flow of events is resumed at {Select Quiz}.

A5: Next

At {Completing Quiz}, if the student clicks Next button,

1. The system shows the next question.
2. The flow of events is resumed at {Completing Quiz}.

A6: Previous

At {Completing Quiz}, if the student clicks Previous button,

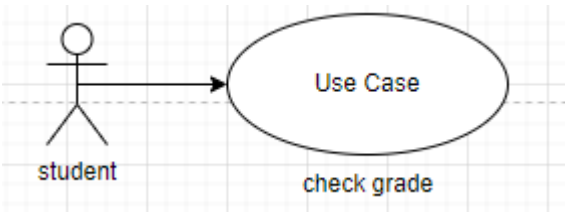
3. The system shows the previous question.
4. The flow of events is resumed at {Completing Quiz}.

Use Case: Check Grade

Brief Description

This use case describes how a student checks their grades for completed quizzes.

Use-case Diagram



Basic Flow

1. The use case begins when the student selects Grade Statistic after logging in the system.

{Displaying Grade}

2. The system displays the grades interface which shows the grade and graph of quizzes the student has done.

{Select Course}

3. The student selects a course to view the grade and clicks Filter.
4. The system retrieves and displays the student's grade and graph for the selected course.

{End Displaying}

5. The use case ends.

Alternative Flows

A1: No Grades Available For Selected Course

At {Select Course}, If there are no grades available to display,

1. The system notifies the student that no grades are available for this course.
2. The flow of events is resumed at {Displaying Grade}.

A2: Reset

At any point between {Displaying Grade} and {End Displaying}, if the student clicks the reset button,

1. The system clears the selected course entry.
2. The flow of events is resumed at {Displaying Grade}.

A3: Refresh

At any point between {Displaying Grade} and {End Displaying}, if the student clicks the refresh button,

1. The system searches new completed quizzes and display them accordingly.
2. The flow of events is resumed at {Displaying Grade}.

Use Case Specification For Teacher Portal

1 Introduction

1.1 Use-Case Name

1. Login
2. Registration
3. Question Bank Management
4. Exam Management

1.2 Brief Description

The teacher portal is the interface for teachers to manage the question bank and exams. The teacher can log in, register, manage the question bank, and manage exams.

2 Actors

The primary actor involved in the use-cases is the **teacher**.

3 Use Cases

3.1 Login

3.1.1 Brief Description

The teacher logs into the system.

3.1.2 Use-Case Diagram

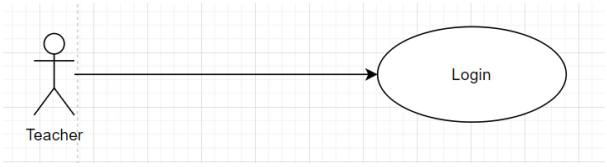


Figure 1: Login Use-Case Diagram

3.1.3 Main Flow

```
1. The teacher enters the initial interface of the system.
2. The teacher enters the username and password.
3. The teacher selects the "login" option.
4. The system validates the username and password.
5. The system displays the main interface.
6. The use case ends.
```

3.1.4 Alternative Flows

A1: Incorrect username or password

```
At {The system validates the username and password}:
1. If the username or password is incorrect, the system displays an error message.
2. The flow of events is resumed at {The teacher enters the username and password}.
```

3.2 Registration

3.2.1 Brief Description

The teacher registers for an account in the system.

3.2.2 Use-Case Diagram

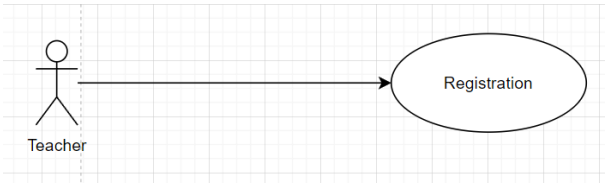


Figure 2: Registration Use-Case Diagram

3.2.3 Main Flow

```
1. The teacher enters the initial interface of the system.
2. The teacher selects the "register" option.

{Registration}
3. While the teacher in the registration interface:
    3.3.1 The system displays the registration form, including the fields:
        - Username
        - Name
        - Gender
        - Age
        - Position
        - Department
        - Password
        - Password confirmation
    {Filling in the registration form}
    3.3.2 The teacher fills in the registration form.
    {Submitting the registration form}
    3.3.3 The teacher selects the "register" option.
    3.3.4 The system stores the teacher's information and displays a success message.
5. The use case ends
```

3.2.4 Alternative Flows

A1: Incomplete form

```
At {Submitting the registration form}:
1. If the form is incomplete, the system displays an error message.
2. The flow of events is resumed at {Filling in the registration form}.
```

A2: Cancel registration

```
At any point between {Registration},
{Filling in the registration form} and
{Submitting the registration form}:
1. If the user selects the "close" option.
2. The system returns to the initial interface.
```

A3: Username already exists

```
At {Submitting the registration form}:
1. The teacher selects the "register" option.
2. If the username already exists, the system displays an error message.
3. The flow of events is resumed at {Filling in the registration form}.
```

A4: Passwords do not match

```
At {Submitting the registration form}:
1. The teacher selects the "register" option.
2. If the passwords do not match, the system displays an error message.
3. The flow of events is resumed at {Filling in the registration form}.
```

3.3 Question Bank Management

3.3.1 Beief Description

The teacher manages the question bank by filtering, adding, updating, and deleting questions.

3.3.2 Use-Case Diagram

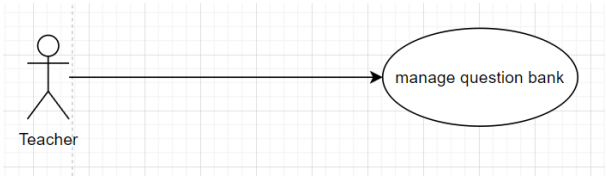


Figure 3: Question Bank Management Use-Case Diagram

3.3.3 Main Flow

```
1. User enters the initial interface of the system.
2. User inputs the correct username and password.
3. User selects the "login" option.
4. The system displays the main interface.
5. User selects the "Question Bank" option.
6. The system displays the question bank interface.

{Select Activity}
7. While the teacher in the question bank interface.
    7.1 If the teacher perform FILTERING QUESTIONS
        {Entering filters for questions}
        7.1.1 The teacher fill in/select the filters for the questions,
        including the "Question", "Type", "Score".
        {Filtering questions}
        7.1.2 The teacher clicks the "Filter" button.
        7.1.3 The system displays the list of questions based on the selected filters.
    7.2 If the teacher perform ADDING A NEW QUESTION
        {Entering a new question}
        7.2.1 The teacher enters the "Question", "Type", "Score" and "Answer".
        {Adding a new question}
        7.2.2 The teacher clicks the "Add" button.
```

```
        7.2.3 The system stores the information of the new question
         and displays a success message.
    7.3 If the teacher perform UPDATING A QUESTION
        {Updating a question}
        7.3.1 The teacher selects a question from the list of questions.
        7.3.2 The system  highlights the selected question.
        7.3.3 The teacher updates the information of the question.
        7.3.4 The teacher clicks the "Update" button.
        7.3.5 The system updates the information of the question
        and displays a success message.
    7.4 If the teacher perform DELETING A QUESTION
        {Deleting a question}
        7.4.1 The teacher selects a question from the list of questions.
        7.4.2 The system  highlights the selected question.
        7.4.3 The teacher clicks the "Delete" button.
        7.4.4 The system deletes the question and
        displays a success message.
8. The use case ends.
```

3.3.4 Alternative Flows

A1: Cancel activity

```
At any point between {Entering filters for questions},
{Filtering questions}, {Entering a new question},
{Adding a new question}, {Updating a question} and {Deleting a question}:
1. The teacher can cancel the activity.
2. The flow of events is resumed at {Select Activity}.
```

A2: Incomplete form of filter questions

```
At {Filtering questions}:
1. If the form is incomplete, the system displays an error message.
2. The flow of events is resumed at {Entering filters for questions}.
```

A3: Incomplete form of adding a new question

```
At {Adding a new question}:
1. If the form is incomplete, the system displays an error message.
2. The flow of events is resumed at {Entering a new question}.
```

A4: Incomplete form of updating a question

```
At {Updating a question}:
1. If the form is incomplete, the system displays an error message.
2. The flow of events is resumed at {Updating a question}.
```

3.4 Exam Management

3.4.1 Brief Description

This use case describes the process of managing exams by the teacher.

3.4.2 Use-Case Diagram

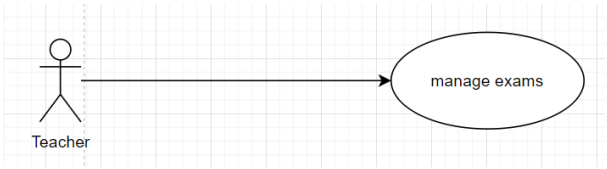


Figure 4: Exam Management Use-Case Diagram

3.4.3 Main Flow

```
1. User enters the initial interface of the system.
2. User inputs the correct username and password.
3. User selects the "login" option.
4. The system displays the main interface.
5. User selects the "Exam" option.
6. The system displays the exam management interface.

{Select Activity}
7. While the teacher has an activity to perform
    7.1 If the teacher perform  FILTER CURRENT EXAM
        {Entering filters for current exams}
        7.1.1 The teacher enters the "Exam Name",
        selects the "CourseID" and select the state of
        "Published/Unpublished" for filtering the current exam.
        {Filtering current exams}
        7.1.2 The teacher clicks the "Filter" button.
        7.1.3 The system retrieves the list of exams based on the
        selected filters and display them in the list of current exams.
    7.2 If the teacher perform FILTERING EXISTING QUESTIONS
        {Entering filters for questions}
        7.2.1 The teacher fill in/select the filters for the questions,
        including the "Question", "Type", "Score".
        {Filtering existing questions}
        7.2.2 The teacher clicks the "Filter" button.
        7.2.3 The system displays the list of questions based on the selected filters.
    7.3 If the teacher perform CREATE NEW EXAM
        {Choosing exam questions}
        7.3.1 The teacher choose the questions from existing questions.
        7.3.2 The system displays the detail of
        {Entering exam details}
        7.3.2 The teacher enters "New Exam Name", "New Exam Time"
        and selects the providing courseID and
        providing option to publish it or not.
        7.3.3 If the teacher press the "Add" button
            {Adding a new exam}
            7.3.3.1 The system adds stores the information
            of the new exam and displays a success message.
        7.3.4 If the teach press the "Update" button
            {Updating an exam}
            7.3.3.1 The system updates the information of the exam
            and displays a success message.
8. The use case ends.
```

3.4.4 Alternative Flows

A1: Cancel activity

```
At any point between {Entering filters for current exams},
{Filtering current exams}, {Entering filters for questions},
```

```
{Filtering existing questions}, {Choosing exam questions},
{Entering exam details}, {Adding a new exam} and {Updating an exam}:
1. The teacher can cancel the activity.
2. The flow of events is resumed at {Select Activity}.
```

A2: Incomplete form of filter current exams

```
At {Filtering current exams}:
1. If the form is incomplete, the system displays an error message.
2. The system displays the exam management interface.
3. The flow of events is resumed at {Entering filters for current exams}.
```

A3: Incomplete form of filter existing questions

```
At {Filtering existing questions}:
1. If the form is incomplete, the system displays an error message.
2. The system displays the exam management interface.
3. The flow of events is resumed at {Entering filters for questions}.
```

A4: Incomplete form of creating a new exam

```
At {Adding a new exam}:
1. If the form is incomplete, the system displays an error message.
2. The system displays the exam management interface.
3. The flow of events is resumed at {Entering exam details}.
```

Comp 3111 Use case Specification of Task 3

Zhu Fei Hao ID:20949355

2024/10/20

Basic flow of grade static for teacher

User select option "teacher login" on the initial interface
{Enter username and password}
User enter username and password
{Invalid username and password}
Teacher interface is shown
User select option "grade static" on the teacher interface
Filter results based on Course, Exam and Students is show
Graphical representation of results, based on Bar chart, Pie chart and Line chart is show
Function buttons : Reset, Filter and Refresh is show
{Select activity}
while Teacher has an activity to perform **do**
    if "Filter" is been clicked **then**
        Read filter parameter
        {Invalid parameter}
        Change content according to the filter parameter
    end if
    if "Refresh" is been clicked **then**
        Refresh content
    end if
    if "Reset" is been clicked **then**
        Reset filter parameter
    end if
end while

Alternative flow of grade static for teacher

A1 : invalid username and password
At **{Invalid username and password}** if the entered username and password is invalid

1: System informs that username and password is invalid
2: The flow of events is resumed at **{Enter username and password}**

A2 : invalid filter parameter
At **{Invalid parameter}** if the entered entered parameter is invalid

1: System informs that parameter is invalid
2: The flow of events is resumed at **{Select activity}**

Basic flow of student management for manager

User select option "manager login" on the initial interface
{Enter username and password}
User enter username and password
{Incorrect username and password}
Manager interface is shown
User select option "Student management" on the manager interface
Student manager interface is shown
Filtered students based on Username, Name and Department is show
Form to add new students based on username, name, age, gender, department and password is show
Function buttons: Reset, Filter, Delete, Refresh, Add and Update is show
{Select activity}
while Manager has an activity to perform **do**
    if "Filter" is been clicked **then**
        Read filter parameter
        {Invalid filter parameter}
        Change content according to the filter parameter
    end if
    if "Delete" is been clicked **then**
        delete selected student locally
    end if
    if "Refresh" is been clicked **then**
        Refresh content
    end if
    if "Add" is been clicked **then**
        Read add parameter
        {Invalid add parameter}
        create student according to parameter locally
    end if
    if "Update" is been clicked **then**
        Update all local change to the system
    end if
    if "Reset" is been clicked **then**
        Reset filter parameter
    end if
end while

Alternative flow of student management for manager

A1 : invalid username and password
At **{Invalid username and password}** if the entered username and password is invalid

1: System informs that username and password is invalid
2: The flow of events is resumed at **{Enter username and password}**

A2 : invalid filter parameter
At **{Invalid filter parameter}** if the entered entered filter parameter is invalid

1: System informs that filter parameter is invalid
2: The flow of events is resumed at **{Select activity}**

A3 : invalid add parameter
At **{Invalid add parameter}** if the entered entered filter parameter is invalid

1: System informs that add parameter is invalid
2: The flow of events is resumed at **{Select activity}**

Basic flow of teacher management for manager

User select option "manager login" on the initial interface
{Enter username and password}
User enter username and password
{Incorrect username and password}
Manager interface is shown
User select option "teacher management" on the manager interface
teacher manager interface is shown
Filtered teacher based on Username, Name and Department is show
Form to add new teacher based on username, name, gender, age, position, department and password is show
Function buttons: Reset, Filter, Delete, Refresh, Add and Update is show
{Select activity}
while Manager has an activity to perform **do**
    if "Filter" is been clicked **then**
        Read filter parameter
        {Invalid filter parameter}
        Change content according to the filter parameter
    end if
    if "Delete" is been clicked **then**
        delete selected teacher locally
    end if
    if "Refresh" is been clicked **then**
        Refresh content
    end if
    if "Add" is been clicked **then**
        Read add parameter
        {Invalid add parameter}
        create teacher according to parameter locally
    end if
    if "Update" is been clicked **then**
        Update all local change to the system
    end if
    if "Reset" is been clicked **then**
        Reset filter parameter
    end if
end while

Alternative flow of teacher management for manager

A1 : invalid username and password
At **{Invalid username and password}** if the entered username and password is invalid

1: System informs that username and password is invalid
2: The flow of events is resumed at **{Enter username and password}**

A2 : invalid filter parameter
At **{Invalid filter parameter}** if the entered entered filter parameter is invalid

1: System informs that filter parameter is invalid
2: The flow of events is resumed at **{Select activity}**

A3 : invalid add parameter
At **{Invalid add parameter}** if the entered entered filter parameter is invalid

1: System informs that add parameter is invalid
2: The flow of events is resumed at **{Select activity}**

Basic flow of course management for manager

User select option "manager login" on the initial interface
{Enter username and password}
User enter username and password
{Incorrect username and password}
Manager interface is shown
User select option "course management" on the manager interface
course manager interface is shown
Filtered course based on Course ID, Course Name and Department
Form to add new course based on Course ID, Course Name and Department is show
Function buttons: Reset, Filter, Delete, Refresh, Add and Update is show
{Select activity}
while Manager has an activity to perform **do**
    if "Filter" is been clicked **then**
        Read filter parameter
        {Invalid filter parameter}
        Change content according to the filter parameter
    end if
    if "Delete" is been clicked **then**
        delete selected course locally
    end if
    if "Refresh" is been clicked **then**
        Refresh content
    end if
    if "Add" is been clicked **then**
        Read add parameter
        {Invalid add parameter}
        create course according to parameter locally
    end if
    if "Update" is been clicked **then**
        Update all local change to the system
    end if
    if "Reset" is been clicked **then**
        Reset filter parameter
    end if
end while

Alternative flow of course management for manager

A1 : invalid username and password
At **{Invalid username and password}** if the entered username and password is invalid

1: System informs that username and password is invalid
2: The flow of events is resumed at **{Enter username and password}**

A2 : invalid filter parameter
At **{Invalid filter parameter}** if the entered entered filter parameter is invalid

1: System informs that filter parameter is invalid
2: The flow of events is resumed at **{Select activity}**

A3 : invalid add parameter
At **{Invalid add parameter}** if the entered entered filter parameter is invalid

1: System informs that add parameter is invalid
2: The flow of events is resumed at **{Select activity}**